Foundations of Computational Mathematics manuscript No.
(will be inserted by the editor)

H. Z. Munthe-Kaas · W. M. Wright

On the Hopf Algebraic Structure of Lie Group Integrators

Received: date / Accepted: date

Abstract A commutative but not cocommutative graded Hopf algebra $\mathcal{H}_N$, based on ordered rooted trees, is studied. This Hopf algebra generalizes the Hopf algebraic structure of unordered rooted trees $\mathcal{H}_C$, developed by Butcher in his study of Runge–Kutta methods and later rediscovered by Connes and Moscovici in the context of non-commutative geometry and by Kreimer where it is used to describe renormalization in quantum field theory. It is shown that $\mathcal{H}_N$ is naturally obtained from a universal object in a category of non-commutative derivations, and in particular, it forms a foundation for the study of numerical integrators based on non-commutative Lie group actions on a manifold. Recursive and non-recursive definitions of the coproduct and the antipode are derived. It is also shown that the dual of $\mathcal{H}_N$ is a Hopf algebra of Grossman and Larson. $\mathcal{H}_N$ contains two well-known Hopf algebras as special cases: The Hopf algebra $\mathcal{H}_C$ of Butcher–Connes–Kreimer is identified as a proper subalgebra of $\mathcal{H}_N$ using the image of a tree symmetrization operator. The Hopf algebra $\mathcal{H}_F$ of the Free Associative Algebra is obtained from $\mathcal{H}_N$ by a quotient construction.

Keywords Hopf algebra · ordered rooted trees · Lie group integrators · Lie–Butcher series · Butcher group · Connes–Kreimer Hopf algebra · Grossman–Larson Hopf algebra

Mathematics Subject Classification (2000) 16W25 · 16W30 · 22E60 · 37M99 · 65L05 · 65L06 · 81R60 · 81T15

H. Z. Munthe-Kaas
Department of Mathematics, University of Bergen, N-5020 Norway
E-mail: H.Munthe-Kaas@mi.uib.no

W. M. Wright
Department of Mathematics, La Trobe University, Victoria 3086, Australia
E-mail: W.Wright@latrobe.edu.au

1 Introduction

Since Cayley [7] in 1857 and 100 years later Merson [23] and followed shortly after by Butcher [5], it has been known that rooted trees are extremely useful for structuring algebras of differential operators and elementary differentials. In 1972 Butcher [6] produced the far-reaching result that Runge–Kutta methods form a group. This was later named the Butcher group in the paper by Hairer and Wanner [17], who made significant contributions to this theory. The Butcher group is defined on the dual of the tree space and it was pointed out by Dür [14] that there exists a one-to-one correspondence between the Butcher group and the commutative graded Hopf algebra of unordered rooted trees. The Hopf algebra of unordered rooted trees has had far-reaching applications in various areas of mathematics and physics. In 1998 a Hopf subalgebra was discovered by Connes and Moscovici [12] during work on an index theorem in non-commutative geometry and by Kreimer [20] in the renormalization method of quantum field theory. Further collaborations between Connes and Kreimer [10,11] have lead to other surprising results; notably a connection with the Riemann–Hilbert problem was established. Brouder [3,4] realized that the mathematical structure of Connes–Kreimer was the same as that of Butcher. Grossman and Larson [16] also developed a cocommutative graded Hopf algebra on a general class of rooted trees. It was shown by Foissy [15] and Hoffman [18] that the commutative Hopf algebra of Butcher and Connes–Kreimer was the dual of the cocommutative Hopf algebra of Grossman–Larson, which corrected the original result of Panaite [34]. Murua [30,31] has developed series expansions of elementary differential operators and shown among other results that the logarithm of such a series is equivalent to the series expansions obtained from backward error analysis.

Recently a great deal of interest has been focused on developing numerical methods which preserve geometric properties of the exact flow. In particular, Lie group integrators, which describe integrators that use Lie group actions on manifolds, were originally proposed by Crouch and Grossman [13] followed shortly after by Lewis and Simo [21,22]. Integrators of this type are now known as Lie group integrators; a survey of these methods is given in [19]. Series expansions for various classes of Lie group methods have been developed; these expansions are generally used to analyze order. Munthe-Kaas [24,25] constructed the order conditions for a special subclass of Lie group methods, where the computations are performed in a Lie algebra, which is a linear space. Later, [26] it was shown that the classical order conditions could be used along with a certain transformation. Owren and Marthinsen [33] developed the general order conditions for the Crouch–Grossman methods with their analysis being based on ordered rooted trees. Recently, Owren [32] derived the order conditions for the commutator free Lie group methods [8], which were derived to overcome some of the problems associated with computing commutators.

In this paper we aim to construct a commutative graded Hopf algebraic structure, which can be used to analyze the order of all Lie group methods. The outline of this paper is as follows: In Section 2 we will introduce ordered trees and forests and describe some useful operations on them. We

will motivate the present Hopf algebra as a universal object in a category of non-commutative derivations, and also briefly discuss Lie–Butcher theory which will be treated in more detail in [29]. In Section 3 we develop the Hopf algebra of ordered trees, giving both recursive and non-recursive definitions of the coproduct and antipode, using certain cutting operations on ordered rooted trees. We show that the Hopf algebra described in this paper is the dual of the Grossman–Larson Hopf algebra, thus generalizing the result of Hoffman [18]. Finally, in Section 4 we use a symmeterization operator to provide an injective Hopf algebra homomorphism from the unordered trees into the ordered trees, establishing the former as a sub-algebra of the latter.

2 Algebras of non-commutative derivations

2.1 An algebra of trees

In this section we will define an algebra N spanned by forests of ordered (and possibly colored) rooted trees. This algebra is a universal ('free') object in a general category of non-commutative derivation algebras, and plays a role in symbolic computing with Lie–Butcher series similar to the role of *free Lie algebras* [28,35] in symbolic computing with Lie algebras.

Let OT denote the set of *ordered colored rooted trees*, and OF denote the (empty and nonempty) words over the alphabet OT, henceforth called the set of empty and non-empty *forests*. It should be noted that, unlike the classical Butcher theory, the ordering of the branches in the trees in OT is important, and likewise the ordering of the trees within the forest OF.

The basic operations involved in building OT and OF are:

- Create the empty forest $\mathbb{I} \in \mathrm{OF}$.
- Create a longer forest from shorter forests by concatenation, $(\omega_1, \omega_2) \mapsto \omega_1\omega_2$.
- Create a tree from a forest by adding a root node, $B^+ : \mathrm{OF} \to \mathrm{OT}$. In the instance where we wish to color the nodes using a set of colors $\mathcal{I}$, we introduce an indexed family of root adding operations $B_i^+ : \mathrm{OF} \to \mathrm{OT}$ for all $i \in \mathcal{I}$. The inverse operation whereby we create a forest from a tree by removing the root node is written $B^- : \mathrm{OT} \to \mathrm{OF}$. This operation extends to OF by $B^-(\omega_1\omega_2) = B^-(\omega_1)B^-(\omega_2)$ and $B^-(\mathbb{I}) = \mathbb{I}$.

The total number of forests, with n nodes colored in i different ways, is defined by modifying the definition of the well known Catalan numbers

$$C_i^n = \frac{i^n}{(n+1)}\binom{2n}{n}, \quad n = 0, 1, 2, \ldots.$$

See A000108 in [36] for various combinatorial representations of the Catalan numbers. For a forest $\omega \in \mathrm{OF}$ we define the *degree*, $\#(\omega)$, as the number of trees in ω as:

$$\begin{aligned}
\#(\mathbb{I}) &= 0,\\
\#(B^+(\omega)) &= 1,\\
\#(\omega_1\omega_2) &= \#(\omega_1) + \#(\omega_2),
\end{aligned}$$

and the *order*, $|\omega|$, as the total number of nodes in all the trees of ω as:

$$
\begin{aligned}
|\mathbb{I}| &= 0, \\
|B^+(\omega)| &= 1 + |\omega|, \\
|\omega_1\omega_2| &= |\omega_1| + |\omega_2|.
\end{aligned}
$$

We let $N = \mathbb{R}\langle \mathrm{OF} \rangle$ denote the linear space of all finite $\mathbb{R}$-linear combinations of elements in OF. This vector space is naturally equipped with an inner product such that all forests are orthogonal,

$$
\langle \omega_1, \omega_2 \rangle = \begin{cases} 1, \text{ if } \omega_1 = \omega_2, \\ 0, \text{ else}, \end{cases} \qquad \text{for all } \omega_1, \omega_2 \in \mathrm{OF}.
$$

For $a \in N$ and $\omega \in \mathrm{OF}$, we let $a(\omega) \in \mathbb{R}$ denote the coefficient of the forest ω, thus a can be written as a sum

$$
a = \sum_{\omega \in \mathrm{OF}} a(\omega)\omega,
$$

where all but a finite number of terms are zero. The space of all infinite sums of this kind is denoted N^* and is the dual space of N, that is

$$
N^* = \{\alpha : N \to \mathbb{R} : \ \alpha \,\text{linear}\}.
$$

We again let $\alpha(a)$ denote the value of $\alpha \in N^*$ on $a \in N$.

The operations B_i^+, B^- and concatenation extend to N by linearity and the distributive law of concatenation, that is

$$
\begin{aligned}
B_i^+(\omega_1 + \omega_2) &= B_i^+(\omega_1) + B_i^+(\omega_2), \\
\omega(\omega_1 + \omega_2) &= \omega\omega_1 + \omega\omega_2.
\end{aligned}
$$

The vector space N with the concatenation product and the grading # forms a graded associative algebra $N = \bigoplus_{j \in \mathbb{Z}} N_j$, where N_j denotes the linear combination of forests with j trees. Alternatively, it is possible to grade this algebra using $|\cdot|$; in this case, N_j denotes the linear combination of forests with the same number of nodes.

Now we introduce a *left grafting product* which has the algebraic structure of a derivation.

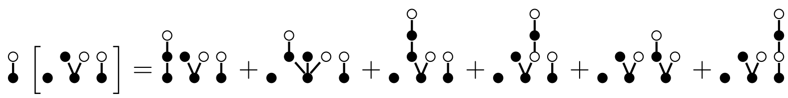
Definition 1 For $\widetilde{a}, a \in N$, define the left grafting $\widetilde{a}[a] \in N$ by the following recursion formulae, where $\tau \in \mathrm{OT}$ and $\omega, \widetilde{\omega} \in \mathrm{OF}$:

$$
\begin{aligned}
\tau[\mathbb{I}] &= 0, \\
\tau[\omega\widetilde{\omega}] &= (\tau[\omega])\widetilde{\omega} + \omega(\tau[\widetilde{\omega}]), \\
\tau[B_i^+(\omega)] &= B_i^+(\tau[\omega]) + B_i^+(\tau\omega),
\end{aligned} \tag{1a}
$$

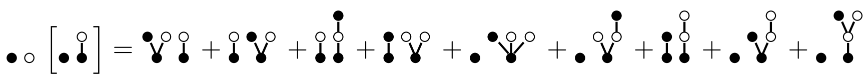
$$
\begin{aligned}
\mathbb{I}[a] &= a, \\
(\tau\omega)[a] &= \tau[\omega[a]] - (\tau[\omega])[a].
\end{aligned} \tag{1b}
$$

The definition of left grafting is extended to the general case $\widetilde{a}[a]$ by bilinearity.

It is useful to understand left grafting directly rather than via the recursive definition. From (1a) we verify that if $\tau \in \mathrm{OT}$ and $\omega \in \mathrm{OF}$ then $\tau[\omega]$ is a sum of $|\omega|$ words, each word obtained by attaching the root of τ with an edge to the left side of a node of ω.



From (1b) we see that if $\tau_1, \tau_2 \in \mathrm{OT}$ and $\omega \in \mathrm{OF}$ then $(\tau_1\tau_2)[\omega]$ is obtained by first left-grafting τ_2 to all nodes of ω and then left-grafting τ_1 to all the nodes of the resulting expression, except to the nodes coming originally from τ_2.



Lemma 1 *If $\tau_1, \ldots, \tau_k \in \mathrm{OT}$ and $\omega \in \mathrm{OF}$ then $(\tau_1 \cdots \tau_k)[\omega]$ is a sum of $|\omega|^k$ words obtained by, in the order $j = k, k-1, \ldots, 1$, attaching the root of the tree τ_j with an edge to the left side of any node in ω. In particular we have*

$$\omega[B_i^+(\mathbb{I})] = B_i^+(\omega), \qquad \textit{for all } \omega \in \mathrm{OF}. \tag{2}$$

Equations (1a) and (1b) imply that for any $d \in N_1$ and $a, b \in N$, we have the Leibniz rule and a composition rule of the form

$$d[ab] = d[a]b + ad[b] \tag{3}$$

$$d[a[b]] = da[b] + d[a][b]. \tag{4}$$

Thus d acts as a first degree derivation on N.

Definition 2 The Grossman–Larson (GL) product $\circ : N \otimes N \to N$ is defined as

$$B_i^+(\omega \circ \widetilde{\omega}) = \omega[B_i^+(\widetilde{\omega})], \qquad \text{for all } \omega, \widetilde{\omega} \in \mathrm{OF},$$

and is extended to the general $a \circ \widetilde{a}$ for $a, \widetilde{a} \in N$ by linearity.

Since $\omega \circ \widetilde{\omega} = B^-\left(\omega[B_i^+(\widetilde{\omega})]\right)$, the GL product can be understood by adding an invisible root to $\widetilde{\omega}$ (turning it into a tree), and left-grafting ω onto all nodes of $B_i^+(\widetilde{\omega})$, including the invisible root. The root is then removed from each of the resulting trees, with the GL product resulting in a total of $(|\widetilde{\omega}| + 1)^{\#(\omega)}$ forests. Some examples of the grafting product from Definition 1 and the GL product from Definition 2 are given in Table 1. In fact, the theory of Grossman and Larson [16] is formulated on trees, not on forests of trees. To a forest ω in our terminology, they add a (proper) root to turn it into a tree. The definition of the GL product in [16] is modified accordingly.

The GL product is an associative #-graded product, $\circ : N_j \otimes N_k \to N_{j+k}$, satisfying for all $a, b, c \in N$, $a \circ \mathbb{I} = \mathbb{I} \circ a$, and

$$\begin{aligned}(a \circ b)[c] &= B^-(a[B_i^+(b)])[c] \\ &= B^-(B_i^+(a[b]) + B_i^+(ab))[c] \\ &= (a[b] + (ab))[c] \\ &= a[b[c]].\end{aligned}$$

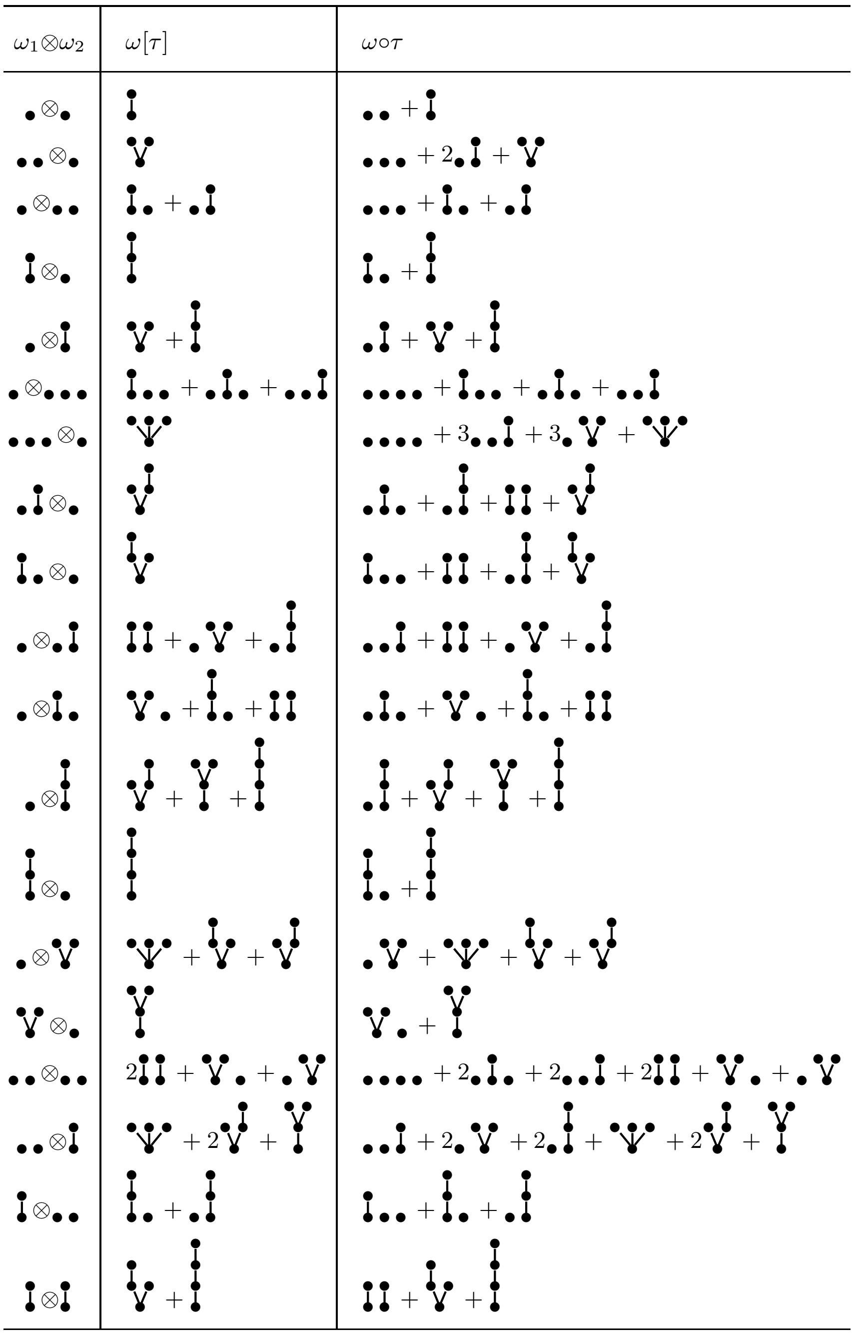


Table 1 The left grafting and Grossman–Larson products for all forests up to order four. The Grossman–Larson product is the dual of the coproduct in $\mathcal{H}_N$ described in Section 3.

2.2 N as a universal object

Definition 3 Let D be an associative $\mathbb{Z}$-graded algebra $D = \bigoplus_{j=0}^{\infty} D_j$ with associative product $a, b \mapsto ab$, a unit $\mathbb{I}$ and grading $\#(D_j) = j$ such that $\#(D_j D_k) = j + k$. We call D a *D-algebra* if it is also equipped with a linear derivation $(\cdot)[\cdot] : D \otimes D \to D$ such that (3) and (4) hold for any $d \in D_1$ and any $a, b \in D$.

Define a D-algebra homomorphism as a linear degree preserving map $\mathcal{F}$ between D-algebras satisfying for any $a, b \in D$:

$$\mathcal{F}(ab) = \mathcal{F}(a)\mathcal{F}(b) \tag{5}$$

$$\mathcal{F}(a[b]) = \mathcal{F}(a)[\mathcal{F}(b)]. \tag{6}$$

Proposition 1 *Let N be the algebra of forests colored with a set $\mathcal{I}$. For any D-algebra D and any map $i \mapsto f_i : \mathcal{I} \to D_1 \subset D$, there exists a unique homomorphism $\mathcal{F} : N \to D$ such that $\mathcal{F}(B_i^+(\mathbb{I})) = f_i$.*

Proof From (2) and (6) we find $\mathcal{F}(B_i^+(\omega)) = \mathcal{F}(\omega)[f_i]$, for any $\omega \in \mathrm{OF}$. Together with (5) and linearity, this shows that by recursion, we can extend $\mathcal{F}$ to a uniquely defined homomorphism defined on all of N. $\square$

This shows that N is a universal object, free over the set $\mathcal{I}$, in the category of D-algebras.

2.3 The algebra of $\mathfrak{G}$ sections on a manifold

As an example of a D-algebra, we consider an algebra related to the numerical Lie group integrators. Let $\mathfrak{g}$ be a Lie algebra of vector fields on a manifold $\mathcal{M}$ and let $\exp : \mathfrak{g} \to \mathrm{Diff}(\mathcal{M})$, denote the flow operator. A basic assumption of numerical Lie group integrators [19,26,33] is the existence of a $\mathfrak{g}$ which is transitive (i.e. spans all tangent directions in any point on $\mathcal{M}$), and for which the exponential map can be computed efficiently and exactly. Transitivity implies that *any* vector field can be written in terms of a function $f : \mathcal{M} \to \mathfrak{g}$. The goal of numerical Lie group integrators is to approximate the flow of a general differential equation

$$y'(t) = f(y)(y), \qquad \text{where } f : \mathcal{M} \to \mathfrak{g}, \tag{7}$$

by composing exponentials of elements in $\mathfrak{g}$. The study of order conditions for Lie group integrators leads to a need for understanding the algebraic structure of non-commuting vector fields on $\mathcal{M}$, generated from f.

Elements $V \in \mathfrak{g}$ are often called invariant or ‘frozen’ vector fields on $\mathcal{M}$. These define first-degree invariant differential operators through the Lie derivative. Let $\mathcal{V}$ be any normed vector space and denote $(\mathcal{M} \to \mathcal{V})$ the set of all smooth functions from $\mathcal{M}$ to $\mathcal{V}$, called the space of $\mathcal{V}$-*sections*. For $V \in \mathfrak{g}$ and $\psi \in (\mathcal{M} \to \mathcal{V})$, the Lie derivative, $V[\psi] \in (\mathcal{M} \to \mathcal{V})$, is defined as

$$V[\psi](p) = \left.\frac{d}{dt}\right|_{t=0} \psi\left(\exp(tV)(p)\right), \qquad \text{for any point } p \in \mathcal{M}.$$

For two elements $V, W \in \mathfrak{g}$ we iterate this definition and define the concatenation VW as the second degree invariant differential operator $VW[\psi] = V[W[\psi]]$. The linear space spanned by the 0-degree identity operator $\mathbb{I}[\psi] = \psi$ and all higher degree invariant derivations is called the *universal enveloping algebra* of $\mathfrak{g}$, denoted $\mathfrak{G}$. This is a graded algebra with the concatenation product and degree $\#(\mathbb{I}) = 0$, $\#(\mathfrak{g}) = 1$ and $\#(VW) = \#(V) + \#(W)$.

Given a norm on the vector space $\mathfrak{G}$, we consider the space of $\mathfrak{G}$ sections[1] $(\mathcal{M} \to \mathfrak{G})$. For two sections $f, g \in (\mathcal{M} \to \mathfrak{G})$ we define $f[g] \in (\mathcal{M} \to \mathfrak{G})$ pointwise from the Lie derivative as

$$f[g](p) = \left(f(p)[g]\right)(p), \qquad p \in \mathcal{M}.$$

Similarly, the concatenation on $\mathfrak{G}$ is extended pointwise to a concatenation $fg \in (\mathcal{M} \to \mathfrak{G})$ as

$$(fg)(p) = f(p)g(p), \qquad p \in \mathcal{M}.$$

From these definitions we find:

Lemma 2 *Let $f \in (\mathcal{M} \to \mathfrak{g})$ and $g, h \in (\mathcal{M} \to \mathfrak{G})$. Then:*

$$\begin{aligned} f[gh] &= f[g]h + gf[h], \\ (f \circ g)[h] \equiv f[g[h]] &= fg[h] + f[g][h]. \end{aligned}$$

Proof For $p \in \mathcal{M}$ let $V = f(p) \in \mathfrak{g}$. Then

$$\begin{aligned} f[gh](p) &= \left.\frac{d}{dt}\right|_{t=0} (gh)(\exp(tV)(p)) \\ &= \left.\frac{d}{dt}\right|_{t=0} g(\exp(tV)(p))h(\exp(tV)(p)) \\ &= \left.\frac{d}{dt}\right|_{t=0} g(\exp(tV)(p))h(p) + g(p)h(\exp(tV)(p)) \\ &= \left(f[g]h + gf[h]\right)(p), \end{aligned}$$

$$\begin{aligned} (f[g[h]])(p) &= \left.\frac{d}{dt}\right|_{t=0} (g(\exp(tf)(p))[h](\exp(tf)(p)) \\ &= \left.\frac{d}{dt}\right|_{t=0} g(p)[h](\exp(tf)(p)) + g(\exp(tf)(p))[h](p) \\ &= (fg)[h](p) + (f[g])[h](p). \end{aligned}$$

$\square$

Note the difference between fg and $f \circ g$. In the concatenation the value of g is frozen to $g(p)$ before the differentiation with f is done, whereas in the latter case the spatial variation of g is seen by the differentiation using f.

[1] Thus $\mathfrak{G}$ is a trivial vector bundle over $\mathcal{M}$, $\mathfrak{g}$ a trivial sub-bundle and the tangent bundle $T\mathcal{M}$ is a non-trivial sub-bundle of $\mathfrak{g}$.

Lemma 2 shows that $(\mathcal{M} \to \mathfrak{G})$ is a D-algebra. Thus if we, for every $i \in \mathcal{I}$, pick a vector field $f_i \in (\mathcal{M} \to \mathfrak{g})$ then there exists a unique homomorphism $\mathcal{F} : N \to (\mathcal{M} \to \mathfrak{G})$ such that $\mathcal{F}(B_i^+(\mathbb{I})) = f_i$. The images of the trees $\mathcal{F}(\tau)$, for $\tau \in \mathrm{OT}$, are called the *elementary differentials* in Butcher's theory (see [5]) and the images of the forests $\mathcal{F}(\omega)$, for $\omega \in \mathrm{OF}$, are called *elementary differential operators* in Merson's theory (see [23]).

2.4 Elements of Lie–Butcher theory

To motivate the algebraic structures of the next section, we briefly introduce some elements of Lie–Butcher theory. This theory is the non-commutative generalization of the classical Butcher theory and is the general foundation behind the construction of order conditions for Lie group integrators. Various aspects of this theory have been developed in [2,24,25,27,33]. A comprehensive treatment is given in [29].

With the vector space N being defined as the finite $\mathbb{R}$-linear combinations of OF, we now let N^* denote the space of infinite $\mathbb{R}$-linear combinations (sequences), or the algebraic dual space of N. All the operations of the previous paragraphs extend to N^* by local finiteness, see [35]. For example, the inner-product in (2.1) is extended to a dual pairing $\langle \cdot, \cdot \rangle : N^* \times N \to \mathbb{R}$, where the computation $\langle \alpha, b \rangle = \sum_{\omega \in \mathrm{OF}} \alpha(\omega) b(\omega)$ is always finite, since b is required to be finite.

Consider the homomorphism $\mathcal{F}$ introduced in Section 2.3 extended to a homomorphism of infinite series $\mathcal{F} \colon N^* \to (\mathcal{M} \to \mathfrak{G})$, where $(\mathcal{M} \to \mathfrak{G})$ should now be understood as a space of formal series. The series might not converge, but all definitions make sense termwise, and any finite truncation yields a proper $\mathfrak{G}$-section. In classical (commutative) Butcher theory the image of $\alpha \in N^*$ is called an S-series, see Murua [30]. Similarly, we define an LS-series as an infinite formal series in $(\mathcal{M} \to \mathfrak{G})$, given by [2]

$$\mathrm{LS}(\alpha) = \sum_{\omega \in \mathrm{OF}} h^{|\omega|} \alpha(\omega) \mathcal{F}(\omega). \tag{8}$$

Classical Lie series on manifolds is a generalization of Taylor series, where the fundamental result is the following 'pull-back formula': Let $f \in (\mathcal{M} \to \mathfrak{g})$, be a vector field and $\exp(f) : \mathcal{M} \to \mathcal{M}$, be the $t = 1$-flow. For any $g \in (\mathcal{M} \to \mathfrak{g})$ we have (see [1]) that

$$g(\exp(f)(p)) = \sum_{j=0}^{\infty} \frac{1}{j!} f^j[g](p) \equiv \mathrm{Exp}(f)[g](p),$$

where $f^0 = \mathbb{I}$ and $f^j[g] = f[\cdots f[f[g]] \cdots] = (f \circ \cdots \circ f)[g]$. Note that if $f = \mathcal{F}(\bullet)$, then $f^j = \mathcal{F}(\bullet \circ \cdots \circ \bullet)$, thus the operator exponential $\mathrm{Exp}(f) = \sum_{j=0}^{\infty} f^j / j!$ is a LS-series.

Two special cases of LS series are of particular importance: A LS-series $\mathrm{LS}(\alpha)$ is called *logarithmic* or *algebra-like* if $\mathrm{LS}(\alpha) \in (\mathcal{M} \to \mathfrak{g})$ represents

[2] See comments at the end of Section 4 on the chosen normalization.

a vector field, and it is is called *exponential* or *group-like* if $\mathrm{LS}(\alpha)$ is the (formal) operator exponential of a logarithmic series. A logarithmic LS-series is the non-commutative generalization of a B-series, named the *Lie–Butcher series* [29].

Note that if $\tau_1, \tau_2 \in \mathrm{OT}$, then $a = \tau_1\tau_2 - \tau_2\tau_1$ is a logarithmic series since it represents the commutator of two vector fields. More generally, a series $\alpha \in N^*$ is logarithmic, if and only if, all its finite components belong to the free Lie algebra generated by OT. A Hall basis for this space is characterized in [27]. Reutenauer [35] presents several alternative characterizations of logarithmic and exponential series.

We find the characterization in terms of *shuffle products* particularly useful. The shuffle product $\sqcup : N \otimes N \to N$ is defined for two forests as the summation over all permutations of the trees in the forests while preserving the ordering of the trees in each of the initial forests, and is extended to $N \otimes N$ by linearity. It can also be recursively defined in the asymmetric way $\mathbb{I} \sqcup \omega = \omega \sqcup \mathbb{I} = \omega$ for any forest $\omega \in \mathrm{OF}$, and if $\omega_1 = \tau_1 \upsilon_1$ and $\omega_2 = \tau_2 \upsilon_2$ for $\tau_1, \tau_2 \in \mathrm{OT}$ and $\upsilon_1, \upsilon_2 \in \mathrm{OF}$, then

$$(\tau_1\upsilon_1) \sqcup (\tau_2\upsilon_2) = \tau_1(\upsilon_1 \sqcup (\tau_2\upsilon_2)) + \tau_2((\tau_1\upsilon_1) \sqcup \upsilon_2).$$

The shuffle product is associative and commutative, for all $\omega_1, \omega_2, \omega_3 \in \mathrm{OF}$ we have

$$\begin{aligned}(\omega_1 \sqcup \omega_2) \sqcup \omega_3 &= \omega_1 \sqcup (\omega_2 \sqcup \omega_3),\\ \omega_1 \sqcup \omega_2 &= \omega_2 \sqcup \omega_1.\end{aligned}$$

Table 2 gives some simple, but nontrivial, examples of the shuffle product.

Lemma 3 *[35] A series $\alpha \in N^*$ is logarithmic if and only if*

$$\begin{aligned}\alpha(\mathbb{I}) &= 0\\ \alpha(\omega_1 \sqcup \omega_2) &= 0 \qquad \textit{for all } \omega_1, \omega_2 \in \mathrm{OF} \setminus \{\mathbb{I}\}.\end{aligned}$$

A series $\alpha \in N^*$ is exponential if and only if

$$\begin{aligned}\alpha(\mathbb{I}) &= 1\\ \alpha(\omega_1 \sqcup \omega_2) &= \alpha(\omega_1)\alpha(\omega_2) \qquad \textit{for all } \omega_1, \omega_2 \in \mathrm{OF}.\end{aligned}$$

The LS-series of an exponential series $\alpha \in N^*$ represents pull-backs, or finite motions on $\mathcal{M}$. They form a group under composition with the Grossman–Larson product, which is the generalization of the *Butcher group* to the case of non-commutative group actions. Consider the GL product as a linear operator $\circ : N^* \otimes N^* \to N^*$ defined by $\circ(\omega_1 \otimes \omega_2) = \omega_1 \circ \omega_2$. To compute the composition $\alpha \circ \beta$ for general $\alpha, \beta \in N^*$, it is useful to introduce the dual of $\circ$, the linear map $\Delta_N : N \to N \otimes N$ to be defined in Section 3.1. Using Corollary 1, we find

$$(\alpha \circ \beta)(\omega) = \langle \circ(\alpha \otimes \beta), \omega \rangle = \langle \alpha \otimes \beta, \Delta_N(\omega) \rangle = \sum_{\omega_1 \otimes \omega_2 \in \Delta_N(\omega)} \alpha(\omega_1)\beta(\omega_2).$$

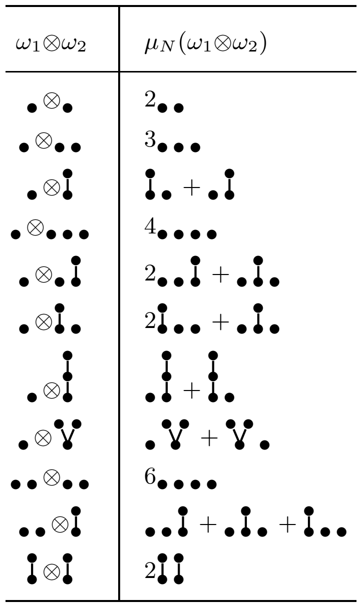
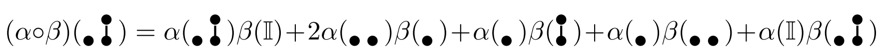


Table 2 All nontrivial examples of the shuffle product for all forests up to and including order four. The shuffle product is the dual of the coproduct in the Grossman–Larson Hopf algebra.

As an illustrative example, we read from Table 5 that



The inverse in the group is found from the *antipode*, a linear map $S_N : N \to N$ defined in Section 3. It can be shown that the dual of the antipode $S_N^* : N^* \to N^*$ also defines the inverse in the group:

$$\alpha^{-1}(\omega) = \alpha(S_N(\omega)) = \langle \alpha, S_N(\omega) \rangle = \langle S_N^*(\alpha), \omega \rangle,$$

for all exponential $\alpha \in N^*$.

3 Hopf algebras

In this section we will study a commutative graded Hopf algebra $\mathcal{H}_N$ of ordered trees. The coproduct Δ_N in the algebra is defined by recursion formulae, and at a later stage we will show that this Δ_N is the dual of the GL product thereby establishing the connection between this Hopf algebra and the algebra of the Butcher group.

We begin by briefly reviewing the definition of a Hopf algebra, see [37] for details. A real associative *algebra* $\mathcal{A}$ is a real vector space with an associative product $\mu : \mathcal{A} \otimes \mathcal{A} \to \mathcal{A}$ and a unit $u : \mathbb{R} \to \mathcal{A}$ such that $\mu(a \otimes u(1)) =$

$\mu(u(1)\otimes a) = a$ for all $a \in \mathcal{A}$. The dual of an algebra is called a *coalgebra*, $\mathcal{C}$, which is a vector space equipped with a coassociative coproduct $\Delta : \mathcal{C} \to \mathcal{C}\otimes\mathcal{C}$ and counit $e : \mathcal{C} \to \mathbb{R}$. A *bialgebra* $\mathcal{B}$ is a linear space which is both an algebra and also a coalgebra structure such that the coproduct and the counit are compatible with the product, in the sense that

$$e(\mu(\omega_1\otimes\omega_2)) = \mu(e(\omega_1)\otimes e(\omega_2)), \tag{9}$$
$$\Delta(\mu(\omega_1\otimes\omega_2)) = (\mu\otimes\mu)(I\otimes T\otimes I)(\Delta(\omega_1)\otimes\Delta(\omega_2)), \tag{10}$$

where $T(\omega_1\otimes\omega_2) = \omega_2\otimes\omega_1$ is the twist map. Let $\mathrm{End}(\mathcal{B})$ denote all linear maps from $\mathcal{B}$ to itself. We define the *convolution* $\star : \mathrm{End}(\mathcal{B}) \to \mathrm{End}(\mathcal{B})$ as

$$(A \star B)(a) = \mu((A\otimes B)\Delta(a)), \qquad \text{for } A, B \in \mathrm{End}(\mathcal{B}) \text{ and } a \in \mathcal{B}. \tag{11}$$

Let $I \in \mathrm{End}(\mathcal{B})$ denote the identity matrix. An *antipode* is a linear map $S \in \mathrm{End}(\mathcal{B})$, which is the two-sided inverse of the identity matrix under convolution, with the antipode satisfying

$$(I \star S)(a) = (S \star I)(a) = u(e(a)), \qquad \text{for all } a \in \mathcal{B}. \tag{12}$$

Definition 4 A Hopf algebra $\mathcal{H}$ is a bialgebra equipped with an antipode.

3.1 The Hopf algebra of ordered trees

We will study a particular Hopf algebra based on the vector space of ordered forests $N = \mathbb{R}\langle \mathrm{OF}\rangle$, where the coproduct is defined by the following recursion.

Definition 5 Let $\Delta_N : N \to N\otimes N$ be defined by linearity and the recursion

$$\begin{aligned}\Delta_N(\mathbb{I}) &= \mathbb{I}\otimes\mathbb{I},\\ \Delta_N(\omega\tau) &= \omega\tau\otimes\mathbb{I} + \Delta_N(\omega) \sqcup\cdot (I\otimes B_i^+)\Delta_N(B^-(\tau)),\end{aligned} \tag{13}$$

where $\tau = B_i^+(\widetilde{\omega}) \in \mathrm{OT}$ and $\omega, \widetilde{\omega} \in \mathrm{OF}$. The linear operation $\sqcup\cdot : N\otimes N\otimes N\otimes N \to N\otimes N$ is a shuffle on the left and concatenation on the right, satisfying

$$(\omega_1\otimes\tau_1) \sqcup\cdot (\omega_2\otimes\tau_2) = (\omega_1 \sqcup \omega_1)\otimes(\tau_1\tau_2).$$

Note that letting $\omega = \mathbb{I}$ yields the special recursion formula for a tree τ:

$$\Delta_N(\tau) = \tau\otimes\mathbb{I} + (I\otimes B_i^+)\Delta_N(B^-(\tau)).$$

Theorem 1 *Let $\mathcal{H}_N$ be the vector space $N = \mathbb{R}\langle \mathrm{OF} \rangle$ with the operations*

$$
\begin{aligned}
&\textit{product}\colon \mu_N(a \otimes b) = a \sqcup b, && \textit{(Shuffle product)}\\
&\textit{coproduct}\colon \Delta_N, && \textit{(Definition 5)}\\
&\textit{unit}\colon u_N(1) = \mathbb{I},\\
&\textit{counit}\colon e_N(\omega) = \begin{cases} 1, \textit{ if } \omega = \mathbb{I},\\ 0, \textit{ else.}\end{cases}
\end{aligned}
$$

Then $\mathcal{H}_N$ is a Hopf algebra with an antipode S_N given by the recursion

$$
\begin{aligned}
&\S_N(\mathbb{I}) = \mathbb{I},\\
&\S_N(\omega\tau) = -\mu_N\left((\S_N \otimes I)\left(\Delta_N(\omega) \sqcup \cdot (I \otimes B_i^+)\Delta_N(B^-(\tau))\right)\right),
\end{aligned}
\tag{14}
$$

where $\tau = B_i^+(\widetilde{\omega}) \in \mathrm{OT}$ and $\omega, \widetilde{\omega} \in \mathrm{OF}$. In particular

$$
\S_N(\tau) = -\mu_N\left((\S_N \otimes I)(I \otimes B_i^+)\Delta_N(B^-(\tau))\right).
$$

Proof The coassociativity of the coalgebra will be established once we have shown that the algebra and coalgebra are compatible. From the fact that $e_N(\omega) = 0$ for all $\omega \in \mathrm{OF} \setminus \{\mathbb{I}\}$ and that the shuffle product of two scalars is just standard multiplication, we immediately get (9). To show (10) we find it convenient to introduce the linear operation $\sqcup\!\sqcup : N \otimes N \otimes N \otimes N \to N \otimes N$ with the shuffle product both on the left and the right, satisfying

$$
(\upsilon_1 \otimes \nu_1) \sqcup\!\sqcup (\upsilon_2 \otimes \nu_2) = (\upsilon_1 \sqcup \upsilon_2) \otimes (\nu_1 \sqcup \nu_2).
$$

The compatibility condition (10) is now equivalent to

$$
\Delta_N(\omega_1 \sqcup \omega_2) = \Delta_N(\omega_1) \sqcup\!\sqcup \Delta_N(\omega_2). \tag{15}
$$

To simplify the notation we use the fact that

$$
\Delta_N(\omega\tau) = \omega\tau \otimes \mathbb{I} + \overline{\Delta}_N(\omega\tau).
$$

Let $\omega_1 = \tau_1\upsilon_1$ and $\omega_2 = \tau_2\upsilon_2$ for $\tau_1, \tau_2 \in \mathrm{OT}$ and $\upsilon_1, \upsilon_2 \in \mathrm{OF}$, now using the recursive definition of the shuffle product and substituting the expression for the coproduct of an ordered forest yields

$$
\begin{aligned}
&\Delta_N(\upsilon_1\tau_1 \sqcup \upsilon_2\tau_2) = \Delta_N((\upsilon_1 \sqcup \tau_2\upsilon_2)\tau_1) + \Delta((\tau_1\upsilon_1 \sqcup u_2)\tau_2)\\
&= (\Delta_N(\upsilon_1) \sqcup\!\sqcup \Delta_N(\upsilon_2\tau_2)) \sqcup \cdot \overline{\Delta}_N(\tau_1) + (\Delta_N(\upsilon_1\tau_1) \sqcup\!\sqcup \Delta_N(\upsilon_2)) \sqcup \cdot \overline{\Delta}_N(\tau_2)\\
&+ (\upsilon_1\tau_1 \sqcup \upsilon_2)\tau_2 \otimes \mathbb{I} + (\upsilon_1 \sqcup \upsilon_2\tau_2)\tau_1 \otimes \mathbb{I}\\
&= (\Delta_N(\upsilon_1) \sqcup\!\sqcup \overline{\Delta}_N(\upsilon_2\tau_2)) \sqcup \cdot \overline{\Delta}_N(\tau_1) + (\Delta_N(\upsilon_1\tau_1) \sqcup\!\sqcup \overline{\Delta}_N(\upsilon_2)) \sqcup \cdot \overline{\Delta}_N(\tau_2)\\
&+ (\Delta_N(\upsilon_1) \sqcup\!\sqcup \upsilon_2\tau_2 \otimes \mathbb{I}) \sqcup \cdot \overline{\Delta}_N(\tau_1) + (\Delta_N(\upsilon_2) \sqcup\!\sqcup \upsilon_1\tau_1 \otimes \mathbb{I}) \sqcup \cdot \overline{\Delta}_N(\tau_2)\\
&+ (\upsilon_1\tau_1 \sqcup \upsilon_2\tau_2) \otimes \mathbb{I}\\
&= \upsilon_1\tau_1 \sqcup \upsilon_2\tau_2 \otimes \mathbb{I} + \overline{\Delta}_N(\upsilon_1\tau_1) \sqcup\!\sqcup (\upsilon_2\tau_2 \otimes \mathbb{I}) + \overline{\Delta}_N(\upsilon_2\tau_2) \sqcup\!\sqcup (\upsilon_1\tau_1 \otimes \mathbb{I})\\
&+ \overline{\Delta}_N(\upsilon_1\tau_1) \sqcup\!\sqcup \overline{\Delta}_N(\upsilon_2\tau_2)\\
&= \Delta_N(\upsilon_1\tau_1) \sqcup\!\sqcup \Delta(\upsilon_2\tau_2).
\end{aligned}
$$

We now have the tools needed to show that the coalgebra is coassociative, which follows from the fact that the coproduct Δ_N satisfies

$$\begin{aligned}(I \otimes \Delta_N)\Delta_N(\omega_1 \sqcup \omega_2) &= (I \otimes \Delta_N)(\Delta_N(\omega_1) \sqcup\!\sqcup\, \Delta_N(\omega_2)) \\ &= (I \otimes \Delta_N)\Delta_N(\omega_1) \sqcup\!\sqcup\, (I \otimes \Delta_N)\Delta_N(\omega_2) \\ &= (\Delta_N \otimes I)\Delta_N(\omega_1) \sqcup\!\sqcup\, (\Delta_N \otimes I)\Delta_N(\omega_2) \\ &= (\Delta_N \otimes I)\Delta_N(\omega_1 \sqcup \omega_2).\end{aligned}$$

Thus we have established the structure of a bialgebra. Substituting Definition 5 in (12) yields the recursion for the antipode (14). □

Both the definition of the coproduct Δ_N and thus the antipode S_N are recursive and difficult to use in practice. To develop non-recursive formulae for these, it is first necessary to define certain cutting operations.

Definition 6 For a given forest $\omega \in \mathrm{OF}$, a *parent* is any node p with at least one branch growing from that node and the *children* are the nodes branching from p. Let p_c denote the number of children of p. Cutting off a child node equates to removing the edge connecting the child to its parent.

- A *nodal left cut* of degree c is a cut where the c leftmost children of a given parent node p are cut off. We can write a nodal left cut as $\ell_p(c)$ where $0 < c \leq p_c$. The cut splits a forest ω into two sub-forests, $P^{\ell_p(c)}(\omega)$ and $R^{\ell_p(c)}(\omega)$, where P is the part cut off, with the forest containing the c leftmost children of p as root nodes and R is the remaining bottom part of ω.
- A *left cut* is a collection of $0 \leq k$ nodal left cuts $\ell = \{\ell_{p_i}(c_i)\}_{i=1}^{k}$, where $\{p_i\}_{i=1}^{k}$ are distinct nodes of ω. This splits ω in k cut-off forests $\{\omega_i\}_{i=1}^{k}$ and a remaining forest $R^{\ell}(\omega)$, where ω_i is the forest containing the nodes connected to the c_i leftmost children of p_i and $R^{\ell}(\omega)$ is the forest of the nodes connected to the original root nodes. We define $P^{\ell}(\omega) \in N$ as

$$P^{\ell}(\omega) = \omega_1 \sqcup \omega_2 \sqcup \cdots \sqcup \omega_k. \tag{16}$$

  Note that the definition of a left cut includes the case $k = 0$, called the *empty cut*, where $R^{\ell}(\omega) = \omega$ and we define $P^{\ell}(\omega) = \mathbb{I}$.
- An *admissible left cut* is a left cut, containing the restriction that any path from a node in ω to the corresponding root is cut no more than once.

We denote by LC, NLC and ALC the set of all left cuts, nodal left cuts and admissible left cuts. To define the coproduct we need to slightly extend the definition of an admissible left cut, which we choose to call a *full admissible left cut*. The full admissible left cuts of $\omega \in \mathrm{OF}$ are obtained by adding an (invisible) root node to form the tree $\tau = B_i^{+}(\omega) \in \mathrm{OT}$, applying an admissible left cut on τ, and finally removing the invisible root node again. We denote by FALC the set of all full admissible left cuts. Thus $\mathrm{FALC}(\omega) = \mathrm{ALC}(\tau)$ and for any $\ell \in \mathrm{FALC}(\omega)$ we have $P^{\ell}(\omega) = P^{\ell}(\tau)$ and $R^{\ell}(\omega) = B^{-}(R^{\ell}(\tau))$. Note that $\mathrm{FALC}(\omega)$ contains the 'cut everything', where $k = 1$, $P^{\ell}(\omega) = \omega$ and $R^{\ell}(\omega) = \mathbb{I}$, as well as the empty cut with $k = 0$, $P^{\ell}(\omega) = \mathbb{I}$

and $R^\ell(\omega) = \omega$. It is useful to note that the order in which the cuts are performed does not affect $P^\ell(\omega)$ or $R^\ell(\omega)$. The order of the cuts is taken care of by the use of the shuffle product in the definition of $P^\ell(\omega)$ given by equation (16).

As an example, we list all the cuttings of an example tree in Table 3.

i	ℓ_i	k	$P^{\ell_i}(\tau)$	$R^{\ell_i}(\tau)$	i	ℓ_i	k	$P^{\ell_i}(\tau)$	$R^{\ell_i}(\tau)$
0		0	$\mathbb{I}$		7		2	$\sqcup \bullet$	$\bullet$
1		1	$\bullet$		8		2	$\sqcup \bullet$	$\bullet$
2		1	$\bullet$		9		2	$\bullet \sqcup \bullet$	$\bullet$
3		1			10		3	$\sqcup \bullet \sqcup \bullet$	$\bullet$
4		1		$\bullet$	11		3	$\bullet \sqcup \bullet\bullet \sqcup \bullet$	$\bullet$
5		2	$\bullet \sqcup \bullet$		12		1		$\mathbb{I}$
6		2	$\bullet\bullet \sqcup \bullet$						

Table 3 The cuts ℓ_i of an example tree τ, where ℓ_{12} is the 'cut everything' full cut. Thus $\mathrm{NLC}(\tau) = \{\ell_1, \ldots, \ell_4\}$, $\mathrm{LC}(\tau) = \{\ell_0, \ldots, \ell_{11}\}$, $\mathrm{ALC}(\tau) = \{\ell_0, \ldots, \ell_5\}$ and $\mathrm{FALC}(\tau) = \{\ell_0, \ldots, \ell_5\} \cup \{\ell_{12}\}$.

Proposition 2 *The coproduct Δ_N of $\mathcal{H}_N$ is non-recursively defined as*

$$\Delta_N(\omega) = \sum_{\ell \in \mathrm{FALC}(\omega)} P^\ell(\omega) \otimes R^\ell(\omega). \tag{17}$$

Proof To prove that the recursive definition (13) and the non-recursive definition (17) of the coproduct are identical, an induction argument on the number of vertices is used. First recall that $\mathrm{FALC}(B^-(\tau)) = \mathrm{ALC}(\tau)$, and for any $\ell \in \mathrm{FALC}(B^-(\tau))$ we have $P^\ell(B^-(\tau)) = P^\ell(\tau)$ and $R^\ell(B^-(\tau)) = B^-(R^\ell(\tau))$, this implies that

$$\sum_{\jmath \in \mathrm{FALC}(B^-(\tau))} P^\jmath(B^-(\tau)) \otimes B_i^+(R^\jmath(B^-(\tau))) = \sum_{\jmath \in \mathrm{ALC}(\tau)} P^\jmath(\tau) \otimes R^\jmath(\tau).$$

Using this fact, the coproduct now takes the form

$$\begin{aligned}
\Delta_N(\omega\tau) &= \omega\tau\otimes\mathbb{I} + \left(\sum_{\ell\in\mathrm{FALC}(\omega)} P^\ell(\omega)\otimes R^\ell(\omega)\right) \sqcup\cdot \left(\sum_{\jmath\in\mathrm{ALC}(\tau)} P^\jmath(\tau)\otimes R^\jmath(\tau)\right)\\
&= \omega\tau\otimes\mathbb{I} + \sum_{\ell\in\mathrm{FALC}(\omega)}\ \sum_{\jmath\in\mathrm{ALC}(\tau)} P^\ell(\omega)\sqcup P^\jmath(\tau)\otimes R^\ell(\omega)R^\jmath(\tau)\\
&= \sum_{\ell\in\mathrm{FALC}(\omega\tau)} P^\ell(\omega\tau)\otimes R^\ell(\omega\tau).
\end{aligned}$$

The last equality is true because the sum over $\ell\in\mathrm{FALC}(\omega)$ and $\jmath\in\mathrm{ALC}(\tau)$ is equivalent to the sum over $\ell\in\mathrm{FALC}(\omega\tau)$ except for the 'cut everything' cut which is equal to the term $\omega\tau\otimes\mathbb{I}$. □

Corollary 1 *The dual of the coproduct Δ_N is $\circ$ the Grossman–Larson product, that is for any $\omega\in N$ and $\omega_1,\omega_2\in N^*$ we have*

$$\langle\omega_1\circ\omega_2,\omega\rangle = \langle\omega_1\otimes\omega_2,\Delta_N(\omega)\rangle.$$

Proof If the sum in (17) had been over ALC instead of FALC, then the dual would have been the left grafting. To see this, we use the characterization of left grafting in Lemma 1, and observe that the nodal left cut corresponds to the dual operation of attaching a number of trees in a given order to a common node, while the shuffles in $P^\ell(\omega)$ corresponds to the dual operation of attaching the forests in all possible ways to different nodes. From Definition 2, we see that when the sum is extended from ALC to FALC, then we obtain the dual of the Grossman–Larson product. □

To present a non-recursive definition of the antipode, we define the reversal map $S_F\colon N\to N$ as

$$\begin{aligned}
S_F(\mathbb{I}) &= \mathbb{I},\\
S_F(\tau_1\tau_2\cdots\tau_j) &= (-1)^j\tau_j\tau_{j-1}\cdots\tau_1, \quad \text{for all } \tau_1\cdots\tau_j\in\mathrm{OF},
\end{aligned} \tag{18}$$

extended to N by linearity. Thus S_F is the unique anti-automorphism of the concatenation algebra which sends $\tau\mapsto-\tau$.

Proposition 3 *The antipode S_N of $\mathcal{H}_N$ is non-recursively defined as*

$$S_N(\omega) = S_F\left(\sum_{\ell\in\mathrm{LC}(\omega)} P^\ell(\omega)\sqcup R^\ell(\omega)\right). \tag{19}$$

Proof In order to prove this result, we need some results about a Hopf algebraic structure of the Free Associative Algebra (FAA) [35]. Given an alphabet A, FAA is the vector space formed by taking all finite linear combinations of words over A. In our case, the alphabet is OT, the words are OF and the vector space is N. A Hopf algebraic structure $\mathcal{H}_F$ is obtained by taking the product $\mu_F=\mu_N$ as the shuffle product and the coproduct Δ_F defined as

the dual of the concatenation product. The antipode is the map S_F defined in (18). We need a characterization of Δ_F and S_F in terms of cutting operations. For a $\omega \in \mathrm{OF}$ let the set of Word Cuts (WC) be a simple cut ℓ which splits a word ω into two parts $\omega_1 = P^\ell(\omega)$ and $\omega_2 = R^\ell(\omega)$ such that $\omega = \omega_1\omega_2$. WC contains both the empty cut where $P^\ell(\omega) = \mathbb{I}$, $R^\ell(\omega) = \omega$ and cut everything where $P^\ell(\omega) = \omega$, $R^\ell(\omega) = \mathbb{I}$. Note that the difference between ALC and FALC is that FALC may contain a nonempty cut from WC. A direct definition of Δ_F is

$$\Delta_F(\omega) = \sum_{\ell \in \mathrm{WC}} P^\ell(\omega) \otimes R^\ell(\omega), \qquad \text{for all } \omega \in \mathrm{OF}.$$

From (12) we find for $\omega \in \mathrm{OF} \setminus \{\mathbb{I}\}$ that

$$0 = (S_F \star I)(\omega) = \mu_F((S_F \otimes I)\Delta_F(\omega)) = \sum_{\ell \in \mathrm{WC}(\omega)} S_F(P^\ell(\omega)) \sqcup R^\ell(\omega).$$

Thus we find a recursive definition of the antipode S_F

$$\begin{aligned} S_F(\mathbb{I}) &= \mathbb{I}, \\ S_F(\omega) &= -\sum_{\ell \in \mathrm{WC}(\omega) \setminus \mathrm{c.e.}} S_F(P^\ell(\omega)) \sqcup R^\ell(\omega), \end{aligned} \tag{20}$$

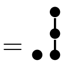
where *c.e.* denotes *cut everything*. Now we repeat the same computation for S_N, using (17). This gives the recursive definition of the antipode S_N

$$\begin{aligned} S_N(\mathbb{I}) &= \mathbb{I}, \\ S_N(\omega) &= -\sum_{\ell \in \mathrm{FALC}(\omega) \setminus \mathrm{c.e.}} S_N(P^\ell(\omega)) \sqcup R^\ell(\omega). \end{aligned} \tag{21}$$

We prove (19) by induction on the number of nodes. Plugging (19) into (21), we find for $\omega \in \mathrm{OF} \setminus \{\mathbb{I}\}$ that

$$\begin{aligned} S_N(\omega) &= -\sum_{\ell \in \mathrm{FALC}(\omega) \setminus \mathrm{c.e.}} S_F\left(\sum_{j \in \mathrm{LC}(P^\ell(\omega))} P^j(P^\ell(\omega)) \sqcup R^j(P^\ell(\omega))\right) \sqcup R^\ell(\omega) \\ &= -\sum_{j \in \mathrm{LC}(\omega)} \sum_{\ell \in \mathrm{WC}(\omega) \setminus \mathrm{c.e.}} S_F\left(P^j(\omega) \sqcup P^\ell(R^j(\omega))\right) \sqcup R^\ell(R^j(\omega)) \\ &= \sum_{j \in \mathrm{LC}(\omega)} S_F\left(P^j(\omega)\right) \sqcup \left(-\sum_{\ell \in \mathrm{WC}(\omega) \setminus \mathrm{c.e.}} S_F(P^\ell(R^j(\omega))) \sqcup R^\ell(R^j(\omega))\right) \\ &= \sum_{j \in \mathrm{LC}(\omega)} S_F\left(P^j(\omega)\right) \sqcup S_F(R^j(\omega)) \\ &= S_F\left(\sum_{j \in \mathrm{LC}(\omega)} P^j(\omega) \sqcup R^j(\omega)\right). \end{aligned}$$

We have used the relation $S_F(\omega_1 \sqcup \omega_2) = S_F(\omega_1) \sqcup S_F(\omega_2)$ see Corollary 2 and the recursion (20), as well as a careful replacement of the summations over $\ell \in \mathrm{FALC}(\omega)$ and $\jmath \in \mathrm{LC}(P^{\ell}(\omega))$ with an equivalent sum over $\jmath \in \mathrm{LC}(\omega)$ and $\ell \in \mathrm{WC}(\omega)$. $\square$

As an example, we compute Δ_N and S_N for the word $\omega =$ . The cuts LC and FALC are shown in Table 4. From the direct formulae we find $\Delta_N(\omega)$ and $\S_N(\omega)$ as listed in Table 5 and Table 6.

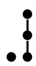
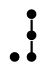
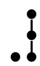
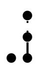
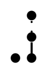
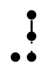
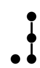
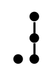
i	ℓ_i	k	$P^{\ell_i}(\omega)$	$R^{\ell_i}(\omega)$	i	ℓ_i	k	$P^{\ell_i}(\omega)$	$R^{\ell_i}(\omega)$
0		0	$\mathbb{I}$		4		1		
1		1			5		2		
2		1			6		2		
3		2			7		1		$\mathbb{I}$

Table 4 Cuts ℓ_i of an example word ω. The cuts $\{\ell_4, \ell_5, \ell_6\}$ are full cuts where the leftmost child of the invisible root is cut and ℓ_7 is the full cut where both the children of the invisible root are cut. Thus $\mathrm{NLC}(\omega) = \{\ell_1, \ell_2\}$, $\mathrm{LC}(\omega) = \{\ell_0, \ldots, \ell_3\}$, $\mathrm{ALC}(\omega) = \{\ell_0, \ldots, \ell_2\}$, $\mathrm{FALC}(\omega) = \{\ell_0, \ldots, \ell_2\} \cup \{\ell_4, \ldots, \ell_7\}$ and $\mathrm{WC}(\omega) = \{\ell_0, \ell_4, \ell_7\}$.

We complete this section by listing some well known but very useful relations of Hopf algebras, see Sweedler [37] for further details.

Corollary 2 *[37] Given $\mathcal{H}_N$ a Hopf algebra, with product $\sqcup$, coproduct Δ_N and antipode S_N, then for all $\omega_1, \omega_2 \in \mathrm{OF}$*

$$(S_N \otimes S_N)\Delta_N(\omega_1) = \Delta_N(S_N(\omega_1)),$$
$$S_N(\omega_1) \sqcup S_N(\omega_2) = S_N(\omega_1 \sqcup \omega_2).$$

If $\mathcal{H}_N$ is either commutative or cocommutative, then $S_N(S_N(\omega)) = \omega$ for all $\omega \in \mathrm{OF}$.

4 Hopf algebras related to $\mathcal{H}_N$

There are two interesting commutative graded Hopf sub-algebras of $\mathcal{H}_N$ obtained by restricting from the set of ordered rooted trees to either the set of tall trees (that is trees where each parent has one child) or bushy trees (that is trees where there is only one parent). These Hopf sub-algebras are useful

respectively for determining the order conditions for the problem (7), when $f(y)$ is constant, or when the numerical scheme has high stage order.

In this section we will also show that the Hopf algebra $\mathcal{H}_C$ of Butcher, based on unordered trees, can be identified as a sub-algebra of $\mathcal{H}_N$. Finally we find that the Hopf algebra $\mathcal{H}_F$ of the Free Associative Algebra is related to $\mathcal{H}_N$ through the operation of *freezing* vector fields, which can be defined as a quotient construction on $\mathcal{H}_N$.

4.1 Connections to the Butcher theory

Let T denote all unordered trees and F denote all unordered forests, defined as the set of all empty or non-empty unordered words over the alphabet T. Recall from [10] the following definition.

Definition 7 Given the real vector space $C = \mathbb{R}\langle \mathrm{F} \rangle$, denote the Hopf algebra of unordered forests as $\mathcal{H}_C = (C, \mu_C, u_C, \Delta_C, e_C, S_C)$. The product $\mu_C : C \otimes C \to C$ is defined as the (commutative) concatenation

$$\mu_C(\omega_1 \otimes \omega_2) = \omega_1 \omega_2.$$

The unit element $u_C : \mathbb{R} \to C$, is given by $u_C(1) = \mathbb{I}$. The coproduct $\Delta_C : C \to C \otimes C$ is defined by linearity and for any $\tau = B_i^+(\widetilde{\omega}) \in \mathrm{T}$ and $\omega, \widetilde{\omega} \in \mathrm{F}$ by the recursion

$$\begin{aligned} \Delta_C(\mathbb{I}) &= \mathbb{I} \otimes \mathbb{I}, \\ \Delta_C(\tau) &= \tau \otimes \mathbb{I} + (I \otimes B_i^+) \Delta_C(B^-(\tau)), \\ \Delta_C(\omega\tau) &= \Delta_C(\omega) \Delta_C(\tau). \end{aligned} \tag{22}$$

The counit $e_C : C \to \mathbb{R}$ is defined by $e_N(\mathbb{I}) = 1$ and $e_N(\omega) = 0$ for $\omega \in \mathrm{OF} \setminus \{\mathbb{I}\}$. The antipode $S_C : C \to C$ is, as usual, the two-sided inverse of the convolution in $\mathcal{H}_C$, see [10] for details.

The main tool used to provide the relationship between the Hopf algebras of ordered and unordered forests is the symmetrization operator defined below.

Definition 8 The symmetrization operator $\Omega : N \to N$ is a map defined by linearity and the relations

$$\begin{aligned} \Omega(\mathbb{I}) &= \mathbb{I}, \\ \Omega(\omega\tau) &= \Omega(\omega) \sqcup \Omega(\tau), \\ \Omega(B_i^+(\omega)) &= B_i^+(\Omega(\omega)). \end{aligned}$$

The shuffle product permutes the trees in a forest in all possible ways, and the symmetrization of a tree is a recursive splitting in sums over all permutations of the branches. The symmetrization defines an equivalence relation on OF, that is, if

$$\Omega(\omega_1) = \Omega(\omega_2) \quad \Longleftrightarrow \quad \omega_1 \sim \omega_2.$$

Thus $\omega_1 \sim \omega_2$ if and only if ω_2 can be obtained from ω_1 by permuting the order of the trees in the forest and the order of the branches of the trees. We see that an alternative characterization of Ω is

$$\Omega(\omega) = \sigma(\omega) \sum_{\substack{\widetilde{\omega} \in \mathrm{OF} \\ \widetilde{\omega} \sim \omega}} \widetilde{\omega}.$$

The integer $\sigma(\omega)$ is the classical symmetry coefficient, defined for trees and forests as

$$\begin{aligned} \sigma(\mathbb{I}) &= 1, \\ \sigma(\tau_1 \tau_2 \cdots \tau_k) &= \sigma(\tau_1) \cdots \sigma(\tau_k) \mu_1! \mu_2! \ldots, \\ \sigma(B_i^+(\tau_1 \cdots \tau_k)) &= \sigma(\tau_1 \cdots \tau_k), \end{aligned}$$

where the integers $\mu_1, \mu_2, \ldots$ count the number of equivalent trees among $\tau_1, \ldots, \tau_k$. In other words, if we consider the full group of all possible permutations of trees and branches acting on a forest $\omega \in \mathrm{OF}$, then $\sigma(\omega)$ is the size of the isotropy subgroup i.e. the number of permutations leaving ω invariant. The total number of permutations acting on a given forest ω is given by the integer $\pi(\omega)$ defined as

$$\begin{aligned} \pi(\mathbb{I}) &= 1, \\ \pi(\tau_1 \tau_2 \cdots \tau_k) &= k! \sigma(\tau_1) \cdots \sigma(\tau_k), \\ \pi(B_i^+(\tau_1 \cdots \tau_k)) &= \pi(\tau_1 \cdots \tau_k). \end{aligned}$$

Note that once a forest ω is symmetrized, then another application of the symmetrization yields the scaling

$$\Omega(\Omega(\omega)) = \pi(\omega) \Omega(\omega). \tag{23}$$

Let F be the unordered forests. Clearly, there is a 1–1 correspondence between unordered forests and equivalence classes of ordered forests, thus there is a natural isomorphism $\mathrm{F} \simeq \mathrm{OF} / \sim$. Through this identification, we can interpret Ω as an injection $\Omega : C \to N$ where $C = \mathbb{R}\langle \mathrm{F} \rangle$ and $N = \mathbb{R}\langle \mathrm{OF} \rangle$. From (23) we see that the map $\Omega^{-1} : N \to C$ defined as

$$\Omega^{-1}(a) = \sum_{\omega \in \mathrm{OF}} \frac{a(\omega)}{\pi(\omega)} \, \mathrm{forget}(\omega),$$

where $\mathrm{forget} : N \to C$ is the natural identification of an ordered forest with the corresponding unordered forest, defines a left-sided inverse $\Omega^{-1}(\Omega(b)) = b$ for all $b \in C$.

Theorem 2 *The symmetrization operator $\Omega : C \to N$ defines an injective Hopf algebra homomorphism from the Hopf algebra $\mathcal{H}_C$ of unordered forests into the Hopf algebra $\mathcal{H}_N$ of ordered forests.*

Proof A Hopf algebra homomorphism is a bialgebra homomorphism, which is a linear map that is both an algebra and a coalgebra homomorphism. Ω is an algebra homomorphism if

$$\begin{aligned}\Omega(u_C(1)) &= u_N(1),\\ \mu_N(\Omega(\omega_1)\otimes\Omega(\omega_2)) &= \Omega(\mu_C(\omega_1\otimes\omega_2)).\end{aligned}$$

These conditions are automatically satisfied by Definition 8. Ω is a coalgebra homomorphism if

$$e_N(\Omega(\omega)) = e_C(\omega), \tag{24}$$

$$\Delta_N(\Omega(\omega)) = (\Omega\otimes\Omega)\Delta_C(\omega). \tag{25}$$

The first condition (24) follows immediately given that the counits are only non-zero when the argument is the empty forest.

The second relation follows using an induction argument. First we need to establish a useful relationship between Ω and Δ_C. Using the *sumless Sweedler notation*

$$\Delta(\omega) = \sum_{\omega_1\otimes\omega_2\in\Delta(\omega)} \omega_1\otimes\omega_2 = \sum_i \omega_{(1)}^{(i)}\otimes\omega_{(2)}^{(i)} \equiv \sum_i \omega_{(1)}\otimes\omega_{(2)} \equiv \omega_{(1)}\otimes\omega_{(2)},$$

we find:

$$\begin{aligned}(\Omega\otimes\Omega)\Delta_C(\omega) \sqcup\!\sqcup (\Omega\otimes\Omega)\Delta_C(\tau) &= (\Omega\otimes\Omega)\left(\omega_{(1)}\otimes\omega_{(2)}\right) \sqcup\!\sqcup (\Omega\otimes\Omega)\left(\tau_{(1)}\otimes\tau_{(2)}\right)\\ &= \Omega(\omega_{(1)})\otimes\Omega(\omega_{(2)}) \sqcup\!\sqcup \Omega(\tau_{(1)})\otimes\Omega(\tau_{(2)})\\ &= \Omega(\omega_{(1)}) \sqcup \Omega(\tau_{(1)})\otimes\Omega(\omega_{(2)}) \sqcup \Omega(\tau_{(2)})\\ &= \Omega(\omega_{(1)}\tau_{(1)})\otimes\Omega(\omega_{(2)}\tau_{(2)})\\ &= (\Omega\otimes\Omega)\omega_{(1)}\tau_{(1)}\otimes\omega_{(2)}\tau_{(2)}\\ &= (\Omega\otimes\Omega)\Delta_C(\omega\tau). \end{aligned}\tag{26}$$

Now we prove (25) by induction. For a forest we find using (15) and (26)

$$\begin{aligned}\Delta_N(\Omega(\omega\tau)) &= \Delta_N(\Omega(\omega)\sqcup\Omega(\tau))\\ &= \Delta_N(\Omega(\omega)) \sqcup\!\sqcup \Delta_N(\Omega(\omega)))\\ &= (\Omega\otimes\Omega)\Delta_C(\omega) \sqcup\!\sqcup (\Omega\otimes\Omega)\Delta_C(\tau))\\ &= (\Omega\otimes\Omega)(\Delta_C(\omega)\Delta_C(\tau))\\ &= (\Omega\otimes\Omega)\Delta_C(\omega\tau).\end{aligned}$$

For a tree $\tau = B_i^+(\omega) \in \mathrm{T}$ and $\widetilde{\omega} \in \mathrm{F}$ we find using the definition of Ω and the recursion formulas (13) and (22) that

$$\begin{aligned}\Delta_N(\Omega(\tau)) &= \Delta_N(\Omega(B_i^+(\omega))) = \Delta_N(B_i^+(\Omega(\omega)))\\ &= B_i^+(\Omega(\omega))\otimes\mathbb{I} + (\mathbb{I}\otimes B_i^+)\Delta_N(\Omega(\omega))\\ &= \Omega(\tau)\otimes\mathbb{I} + (\mathbb{I}\otimes B_i^+)(\Omega\otimes\Omega)\Delta_C(\omega)\\ &= (\Omega\otimes\Omega)(\tau\otimes\mathbb{I} + (\mathbb{I}\otimes B_i^+)\Delta_C(\omega))\\ &= (\Omega\otimes\Omega)\Delta_C(\tau).\end{aligned}$$

The proof extends to a general $a \in N$ by linearity. $\square$

A consequence of the above theorem is a connection between the antipodes of the corresponding Hopf algebras.

Corollary 3 *Given $\Omega : C \to N$ is a bialgebra homomorphism then for $\omega \in$ F, it follows that*

$$S_N(\Omega(\omega)) = \Omega(S_C(\omega)).$$

Note that symmetrization operator Ω is invertible, so expressions for the product, coproduct and antipode of the Hopf algebra of unordered forests can be directly expressed in terms of the corresponding functions in the Hopf algebra of ordered forests, they are

$$\begin{aligned}\mu_C(\omega_1 \otimes \omega_2) &= \Omega^{-1}(\mu_N(\Omega(\omega_1)\otimes\Omega(\omega_2))),\\ \Delta_C(\omega) &= (\Omega\otimes\Omega)^{-1}\Delta_N(\Omega(\omega)),\\ S_C(\omega) &= \Omega^{-1}(S_N(\Omega(\omega))).\end{aligned}$$

In the final part of this section we will elaborate on the connections between the LS-series built on ordered trees, and their commutative counterpart the S-series. These series belong respectively to the dual spaces N^* and C^*, are naturally associated through the dual map $\Omega^* : N^* \to C^*$ taking the series of ordered forests to unordered forests. If $\alpha \in N^*$ and $\beta = \Omega^*(\alpha) \in C^*$, we find

$$\beta(\omega) = \langle \Omega^*(\alpha), \omega\rangle = \langle \alpha, \Omega(\omega)\rangle = \sigma(\omega) \sum_{\widetilde{\omega}\sim\omega} \alpha(\widetilde{\omega}).$$

On a manifold with a commutative Lie group action the elementary differential operators $\mathcal{F}(\omega)$ do not depend on the ordering. Thus we find that the S-series of β as defined in [30] equals the LS-series of α as given in (8),

$$\mathrm{LS}(\alpha) = \sum_{\omega\in\mathrm{OF}} h^{|\omega|}\alpha(\omega)\mathcal{F}(\omega) = \sum_{\omega\in\mathrm{F}} \frac{h^{|\omega|}\beta(\omega)}{\sigma(\omega)}\mathcal{F}(\omega) = S(\beta).$$

This shows that the normalization $1/\sigma(\omega)$ in the commutative case is compatible with our normalization in the non-commutative case.

It is interesting to characterize the image of the *logarithmic* and *exponential* series under Ω^*. If α is logarithmic (Lemma 3) then

$$\begin{aligned}\beta(\mathbb{I}) &= \alpha(\mathbb{I}) = 0,\\ \beta(\omega\tau) &= \langle \alpha, \Omega(\omega\tau)\rangle = \langle \alpha, \Omega(\omega) \sqcup \Omega(\tau)\rangle = 0, \quad \text{for } \omega, \tau \neq \mathbb{I},\end{aligned}$$

thus β is non-zero only on trees. If α is exponential then

$$\begin{aligned}\beta(\mathbb{I}) &= \alpha(\mathbb{I}) = 1,\\ \beta(\tau_1\cdots\tau_k) &= \langle \alpha, \Omega(\tau_1\cdots\tau_k)\rangle = \langle \alpha, \Omega(\tau_1) \sqcup \cdots \sqcup \Omega(\tau_k)\rangle\\ &= \alpha(\Omega(\tau_1))\cdots\alpha(\Omega(\tau_k)) = \beta(\tau_1)\cdots\beta(\tau_k).\end{aligned}$$

This is a well known condition in the composition of B-series, see [9].

4.2 The Free Associative Algebra and frozen vector fields

In the proof of Proposition 3, we defined the Hopf algebraic structure $\mathcal{H}_F$ as the Free Associative Algebra built of words over an alphabet A, where μ_F is the shuffle and Δ_F the dual of the concatenation product. We will briefly comment upon the connection between $\mathcal{H}_N$ and $\mathcal{H}_F$ in the context of Lie group integrators.

In the theory of Lie group integrators, it is common to call constant sections ($\mathcal{M} \to \mathfrak{g}$) *frozen vector fields*. If f is frozen, then the Lie derivative $g[f] = 0$ for all vector fields g. On the algebraic side, a tree $\tau \in \mathrm{OT}$ represents a frozen vector field if the left grafting of anything non-constant to the tree is zero,

$$\widetilde{\tau}[\tau] = 0, \qquad \text{for all } \widetilde{\tau} \in \mathrm{OT} \setminus \{\mathbb{I}\}.$$

In this case we see that the Grossman–Larson product becomes just the concatenation product $\widetilde{\tau} \circ \tau = \widetilde{\tau}\tau$. The freezing of certain vector fields can be understood as the quotient $\mathcal{H}_F = \mathcal{H}_N / G$, where G is the linear span of any $\widetilde{\tau} \in \mathrm{OT} \setminus \{\mathbb{I}\}$ grafted to a frozen vector field. As a special example, we consider the case where all single-node trees are frozen, so that taller trees cannot be produced. Letting $A = \{B_i^+(\mathbb{I})\}_{i \in \mathcal{I}} = \{a_i\}_{i \in \mathcal{I}}$ be the alphabet of all single-node trees, we find from (13) the following well known recursion for the coproduct

$$\begin{aligned} \Delta_F(\mathbb{I}) &= \mathbb{I} \otimes \mathbb{I}, \\ \Delta_F(\omega a_i) &= \omega a_i \otimes \mathbb{I} + \Delta_F(\omega) \otimes (\mathbb{I} \otimes a_i). \end{aligned}$$

5 Concluding remarks

We have in this paper investigated the algebraic structure of the Hopf algebra underlying numerical Lie group integrators. We have developed both recursive and direct formulae for the coproduct and the antipode, and we have in particular emphasized the connection to the Hopf algebra of classical Butcher theory and to the free associative Hopf algebra. We believe that this work is of particular interest for the construction of symbolic software packages dealing with computations involving algebras of non-commutative derivations. The algebraic structure of $\mathcal{H}_N$ is of a universal nature and should be of interest also outside the field of numerical integration, for example, in the renormalization of quantum field theory and the Chen–Fliess theory for optimal control.

References

1. R. Abraham, J. E. Marsden, and T. Ratiu. *Manifolds, Tensor Analysis, and Applications*. AMS 75. Springer-Verlag, Second edition, 1988.
2. H. Berland and B. Owren. Algebraic structures on ordered rooted trees and their significance to Lie group integrators. In *Group theory and numerical analysis*, volume 39 of *CRM Proc. Lecture Notes*, pages 49–63. Amer. Math. Soc., 2005.

3. Ch. Brouder. Runge–Kutta methods and renormalization. *Eur. Phys. J. C*, 12:521–534, 2000.
4. Ch. Brouder. Trees, renormalization and differential equations. *BIT*, 44(3):425–438, 2004.
5. J. C. Butcher. Coefficients for the study of Runge-Kutta integration processes. *J. Austral. Math. Soc.*, 3:185–201, 1963.
6. J. C. Butcher. An algebraic theory of integration methods. *Math. Comp.*, 26:79–106, 1972.
7. A. Cayley. On the Theory of Analytic Forms Called Trees. *Philos. Mag.*, 13:19–30, 1857.
8. E. Celledoni, A. Marthinsen, and B. Owren. Commutator-free Lie group methods. *FGCS*, 19(3):341–352, 2003.
9. P. Chartier, E. Faou, and A. Murua. An algebraic approach to invariant preserving integrators: The case of quadratic and hamiltonian invariants. Submitted, 2005.
10. A. Connes and D. Kreimer. Hopf algebras, renormalization and noncommutative geometry. *Comm. Math. Phys.*, 199(1):203–242, 1998, arXiv:hep-th/9808042.
11. A. Connes and D. Kreimer. Renormalization in quantum field theory and the Riemann–Hilbert problem. I. The Hopf algebra structure of graphs and the main theorem. *Comm. Math. Phys.*, 210(1):249–273, 2000, arXiv:hep-th/9912092.
12. A. Connes and H. Moscovici. Hopf algebras, cyclic cohomology and the transverse index theorem. *Comm. Math. Phys.*, 198(1):199–246, 1998, arXiv:math.DG/9806109.
13. P. E. Crouch and R. Grossman. Numerical integration of ordinary differential equations on manifolds. *J. Nonlinear Sci.*, 3:1–33, 1993.
14. Arne Dür. *Möbius functions, incidence algebras and power series representations*, volume 1202 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 1986.
15. L. Foissy. Les algèbres de Hopf des arbres enracinés décorés. II. *Bull. Sci. Math.*, 126(4):249–288, 2002.
16. R. Grossman and R. G. Larson. Hopf-algebraic structure of families of trees. *J. Algebra*, 126(1):184–210, 1989.
17. E. Hairer and G. Wanner. On the Butcher group and general multi-value methods. *Computing (Arch. Elektron. Rechnen)*, 13(1):1–15, 1974.
18. M. E. Hoffman. Combinatorics of rooted trees and Hopf algebras. *Trans. Amer. Math. Soc.*, 355(9):3795–3811, 2003.
19. A. Iserles, H. Z. Munthe-Kaas, S. P. Nørsett, and A. Zanna. Lie-group methods. *Acta Numerica*, 9:215–365, 2000.
20. D. Kreimer. On the Hopf algebra structure of perturbative quantum field theories. *Adv. Theor. Math. Phys.*, 2:303–334, 1998.
21. D. Lewis and J. C. Simo. Conserving algorithms for the dynamics of Hamiltonian systems of Lie groups. *J. Nonlinear Sci.*, 4:253–299, 1994.
22. D. Lewis and J. C. Simo. Conserving algorithms for the n dimensional rigid body. *Fields Inst. Com.*, 10, 1995.
23. R. H. Merson. An operational method for the study of integration processes. In *Proc. Conf., Data Processing & Automatic Computing Machines*, pages 110–1–110–25, 1957.
24. H. Z. Munthe-Kaas. Lie–Butcher theory for Runge–Kutta methods. *BIT*, 35(4):572–587, 1995.
25. H. Z. Munthe-Kaas. Runge–Kutta methods on Lie groups. *BIT*, 38(1):92–111, 1998.
26. H. Z. Munthe-Kaas. High order Runge–Kutta methods on manifolds. In *Proceedings of the NSF/CBMS Regional Conference on Numerical Analysis of Hamiltonian Differential Equations (Golden, CO, 1997)*, volume 29,1, pages 115–127, 1999.
27. H. Z. Munthe-Kaas and S. Krogstad. On enumeration problems in Lie–Butcher theory. *FGCS*, 19(7):1197–1205, 2003.

28. H. Z. Munthe-Kaas and B. Owren. Computations in a free Lie algebra. *R. Soc. Lond. Philos. Trans. Ser. A Math. Phys. Eng. Sci.*, 357(1754):957–981, 1999.
29. H. Z. Munthe-Kaas and W. M. Wright. Lie–butcher theory. In preperation, 2006.
30. A. Murua. Formal series and numerical integrators. I. Systems of ODEs and symplectic integrators. *Appl. Numer. Math.*, 29(2):221–251, 1999.
31. A. Murua. The Hopf algebra of rooted trees, free Lie algebras, and Lie series. Accepted, 2005.
32. B. Owren. Order conditions for commutator-free lie group methods. Submitted, 2005.
33. B. Owren and A. Marthinsen. Runge–Kutta methods adapted to manifolds and based on rigid frames. *BIT*, 39(1):116–142, 1999.
34. F. Panaite. Relating the Connes–Kreimer and Grossman–Larson Hopf algebras built on rooted trees. *Lett. Math. Phys.*, 51(3):211–219, 2000, arXiv:math.QA/0003074.
35. Ch. Reutenauer. *Free Lie Algebras*. Oxford University Press, 1993.
36. N. J. A. Sloane. The on-line encyclopedia of integer sequences. *Notices Amer. Math. Soc.*, 50(8):912–915, 2003.
37. M. E. Sweedler. *Hopf algebras*. Mathematics Lecture Note Series. W. A. Benjamin, Inc., New York, 1969.

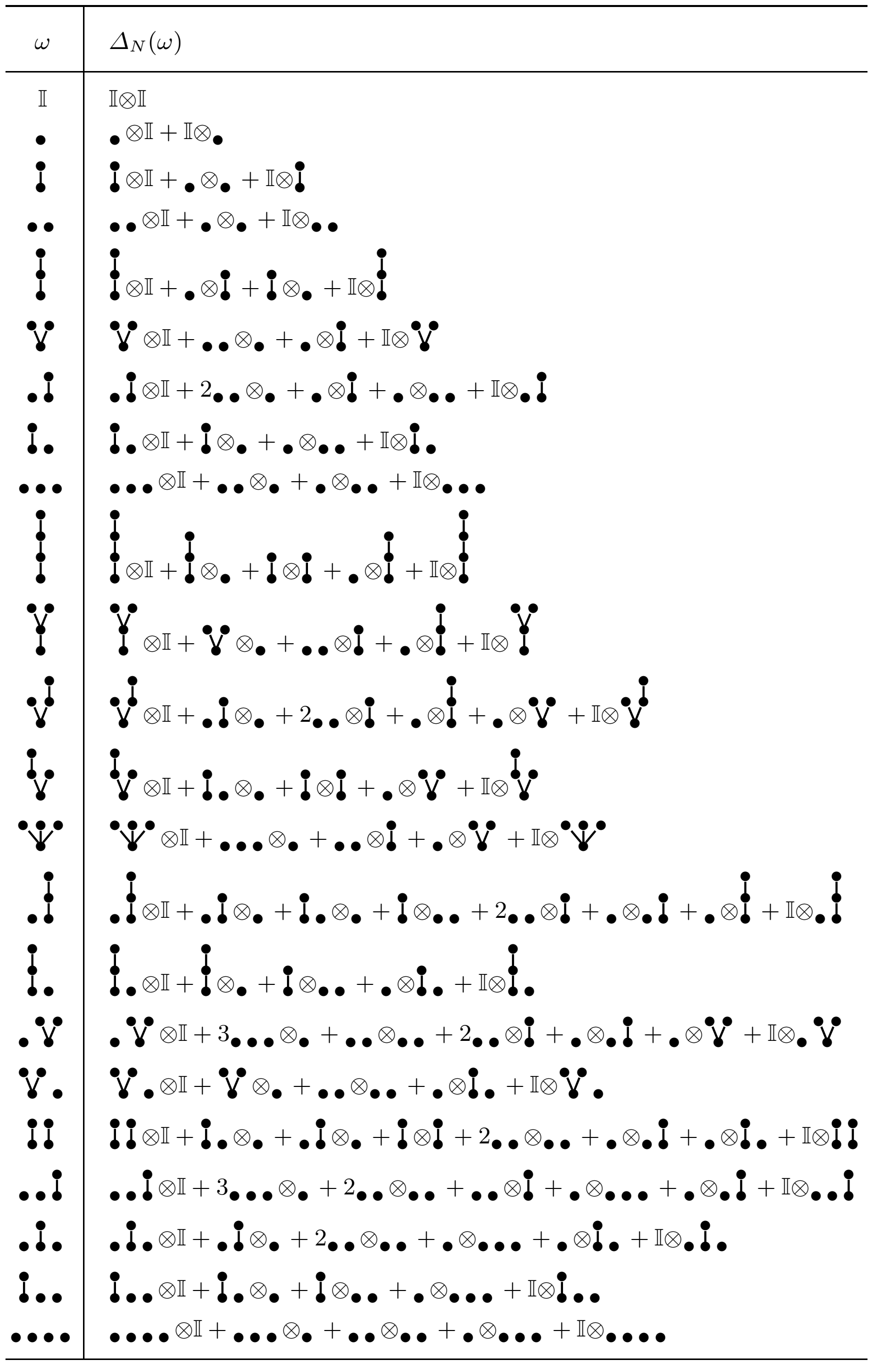


Table 5 The coproduct in $\mathcal{H}_N$ for all forest up to and including order four. This is the dual of the product in the Grossman–Larson Hopf algebra.

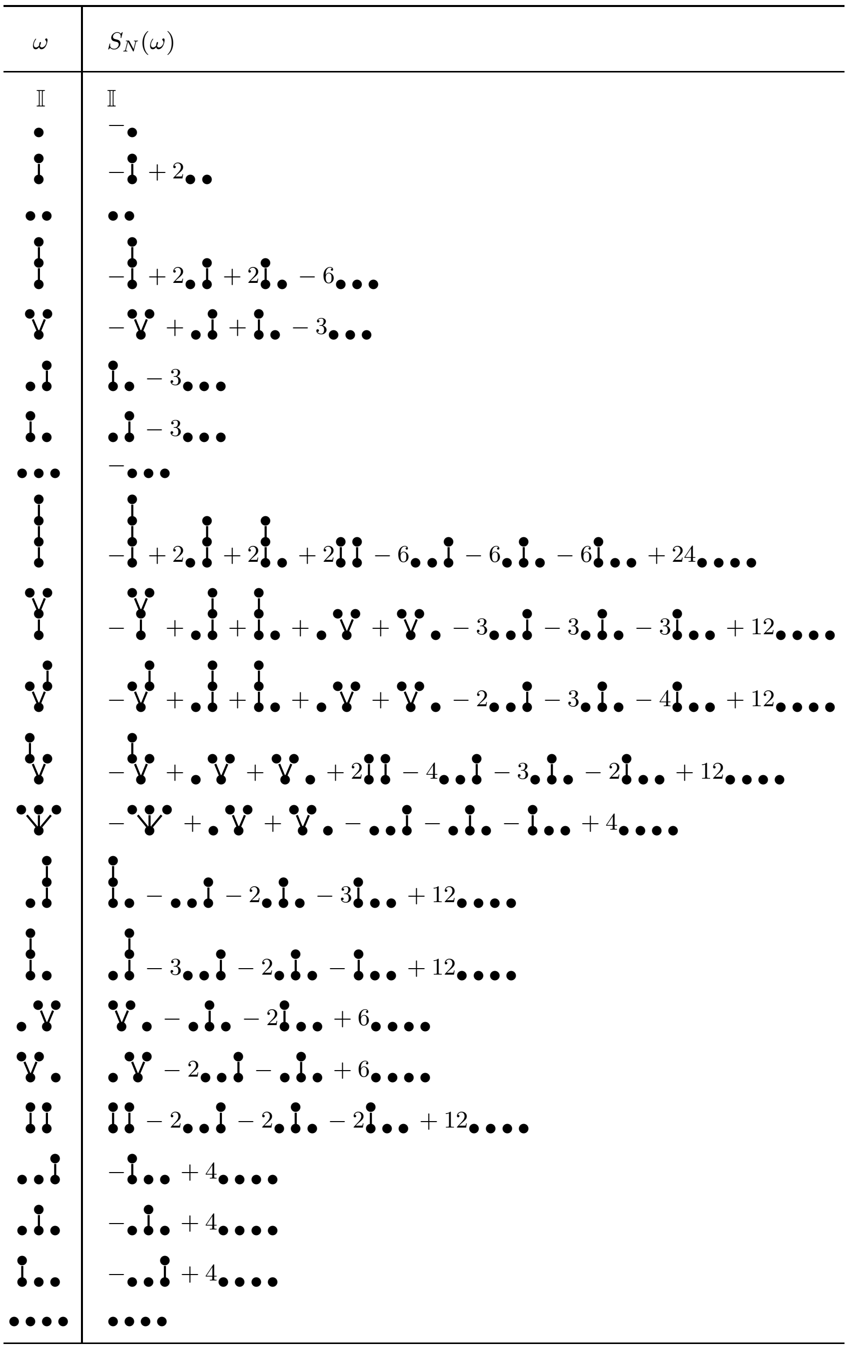


Table 6 The antipode in $\mathcal{H}_N$ for all forests up to and including order four. This is the dual of the Grossman–Larson antipode.